

# **PROCEDE ET SYSTEME POUR GERER DES EXECUTABLES A BIBLIOTHEQUES PARTAGEES**

**Publication number:** FR2820221 (A1)

**Publication date:** 2002-08-02

**Inventor(s):** VERTES MARC PHILIPPE +

**Applicant(s):** CIMAI TECHNOLOGY [FR] +

**Classification:**

- International: G06F9/42; G06F9/445; G06F9/54; G06F9/40; G06F9/445;

G06F9/46; (IPC1-7): G06F17/30; G06F9/40

- European: G06F9/42; G06F9/445L

**Application number:** FR20010001378 20010201

**Priority number(s):** FR20010001378 20010201

**Also published as:**

FR2820221 (B1)

US2004111720 (A1)

US7506316 (B2)

JP2004530184 (T)

WO02061579 (A2)

more >>

**Cited documents:**

US6154878 (A)

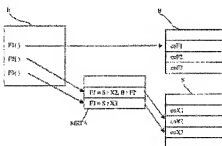
US5175828 (A)

US5495612 (A)

XP000913803 (A)

**Abstract of FR 2820221 (A1)**

The invention relates to a method and system for non-intrusively modifying an executable file E comprising references (F1, F2 and F3) with functions disposed in at least one shared library B within a computer. According to the invention, at least two new files are created, a META file comprising several unresolved references (F2, F3) in executable file E and a services file comprising new functions or services (X2, X3) to be inserted in executable file E. The link editing phase assigns the unresolved references to the META file as a priority and subsequently to shared library B. When said references are listed in the META file, a link is then established for said reference between executable file E and the META file in the place of shared library B; it is then possible, in the META file, to assign one or more functions stored in service file S and/or in shared library B to said reference. The invention can be used for the non-intrusive incremental extension of operating systems. It is no longer necessary to perform modifications inside the nucleus in order to modify the performance of the system.



Data supplied from the [espacenet](http://www.espacenet.com) database — Worldwide

12

DEMANDE DE BREVET D'INVENTION

A1

22 Date de dépôt : 01.02.01.

30 Priorité :

43 Date de mise à la disposition du public de la  
demande : 02.08.02 Bulletin 02/31.

58 Liste des documents cités dans le rapport de  
recherche préliminaire : Ce dernier n'a pas été  
établi à la date de publication de la demande.

60 Références à d'autres documents nationaux  
apparentés :

71 Demandeur(s) : CIMAI TECHNOLOGY Société ano-  
nyme — FR.

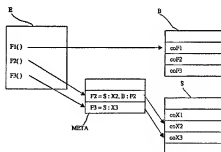
72 Inventeur(s) : VERTES MARC PHILIPPE.

73 Titulaire(s) :

74 Mandataire(s) : PONTET ET ALLANO SARL.

64 PROCÉDE ET SYSTEME POUR GERER DES EXECUTABLES A BIBLIOTHEQUES PARTAGEES.

57 L'invention concerne un procédé et un système pour modifier de manière non intrusive un fichier exécutable E comprenant des références (F1, F2 et F3) à des fonctions disposées dans au moins une bibliothèque partagée B au sein d'un ordinateur. Selon l'invention on crée au moins deux nouveaux fichiers, un fichier META comprenant quelques références non résolues (F2, F3) dans le fichier exécutable E et un fichier de services comprenant de nouvelles fonctions ou services (X2, X3) à insérer dans le fichier exécutable E. La phase d'édition de liens affecte en priorité au fichier META puis à la bibliothèque partagée B les références non résolues. Lorsque ces références sont répertoriées dans le fichier META, alors on établit, pour cette référence, un lien entre le fichier exécutable E et le fichier META à la place de la bibliothèque partagée B. Dans le fichier META on a ensuite la possibilité d'associer à cette référence une ou plusieurs fonctions stockées dans le fichier de service S et/ou dans la bibliothèque partagée B. L'invention permet l'extension incrémentale et non intrusive des systèmes d'exploitation. Il n'est plus nécessaire d'effectuer des modifications au coeur du noyau pour modifier le comportement du système.



"Procédé et système pour gérer des exécutable  
à bibliothèques partagées."

5

La présente invention concerne un procédé et un système pour gérer des fichiers exécutable utilisant des bibliothèques partagées.

10 D'une façon générale, pour éviter des fichiers exécutable de grande capacité, on rassemble dans un fichier distinct l'ensemble de fonctions communes à ces fichiers exécutable. Ce fichier distinct ou bibliothèque partagée est habituellement intégré dans des systèmes d'exploitation. Ainsi, les codes objets définissant une fonction ne sont plus intégrés dans des fichiers exécutable, seule la référence à cette fonction est indiquée dans les fichiers exécutable.

15 Un fichier exécutable utilisant au moins une bibliothèque partagée, ou fichier exécutable dynamique, fait donc appel à des fonctions d'usage commun contenues dans la bibliothèque partagée. Si l'on désire modifier même une partie mineure du fichier exécutable, on est obligé d'effectuer la modification sur l'ensemble du code source du fichier exécutable, puis de  
20 réaliser à nouveau une phase de compilation de ce code source. Cependant, le code source d'un fichier exécutable n'est pas toujours disponible.

Le but de l'invention est de permettre une modification dynamique du comportement d'un fichier exécutable dynamique de manière non intrusive,  
25 c'est à dire sans modification physique du fichier d'exécutable et d'éventuelles bibliothèques partagées.

La présente invention a également pour but d'étendre, de manière dynamique et non intrusive, les capacités des fichiers exécutable existants durant leur exécution.

30 On atteint les objectifs ci-avant avec un procédé pour modifier de manière non intrusive un fichier exécutable comprenant des références à

des fonctions disposées dans au moins une bibliothèque partagée au sein d'un ordinateur. Selon l'invention :

- au lancement du fichier exécutable, on pré-charge un fichier de référence comprenant une pluralité de références à des fonctions nouvelles disposées dans un fichier de services,
- lors de la phase d'édition dynamique de liens pendant le lancement du fichier exécutable, pour chaque référence non résolue comprise dans le fichier exécutable, on affecte en priorité ladite référence au fichier de référence dans la mesure où cette référence est définie dans ce fichier de référence, sinon on affecte cette référence à la bibliothèque partagée,
- au cours de l'exécution du fichier exécutable, l'appel à une fonction référencée dans le fichier de référence, permet l'exécution d'une fonction nouvelle disposée dans le fichier de service au lieu de l'exécution de la fonction réellement appelée, le chemin d'accès à cette fonction nouvelle étant définie dans le fichier de référence.

Une référence est considérée non résolue lorsque le code objet de la fonction à laquelle elle renvoie n'est pas inclus dans le fichier exécutable.

- Un fichier de services est un fichier renfermant de nouveaux services ou fonctions que l'on souhaite adjoindre au fichier exécutable. Le fichier de références comprend une partie des références présentes dans le fichier exécutable. D'une façon conventionnelle, ces références ont pour but l'exécution des fonctions de la bibliothèque partagée. Mais avec le procédé selon l'invention, avant de faire le lien entre chaque référence du fichier exécutable et une fonction correspondante dans la bibliothèque partagée, on vérifie d'abord si cette fonction est référencée dans le fichier de références. Dans l'affirmatif, on dit que le fichier de références a intercepté cette fonction correspondante et on réalise alors un lien entre cette référence et le fichier des références. Dans le cas contraire, c'est-à-dire la fonction n'est pas référencée dans le fichier de références, on effectue de

façon conventionnelle un lien dynamique entre le fichier exécutable et la fonction correspondante dans la bibliothèque partagée.

Lorsqu'une fonction est interceptée, le fichier de référence comprend une référence à cette fonction, et on associe une ou plusieurs fonctions nouvelles à cette référence. En d'autres termes, à chaque référence du fichier de références est associé un chemin d'accès vers une nouvelle fonction. Cette nouvelle fonction peut être une fonction (ou service) présente dans le fichier de services. Par ailleurs, le chemin d'accès défini dans le fichier de référence peut en outre permettre l'exécution d'une fonction de la bibliothèque partagée. Le fait d'intercepter une fonction permet donc de substituer cette fonction originale soit par de nouvelles fonctions, soit par une combinaison de fonctions comprenant ou non la fonction originale stockée dans la bibliothèque partagée.

Dans un mode particulier de réalisation, lors du premier appel à une fonction référencée dans le fichier de référence, on effectue une nouvelle phase d'édition dynamique de liens entre les références comprises dans le fichier de référence et des fonctions en relation avec ces références. On peut effectuer cette nouvelle phase d'édition dynamique de liens en utilisant des routines de manipulation de bibliothèques partagées.

Avantageusement, on peut donc gérer le fichier de référence et le fichier de services en tant que des bibliothèques partagées.

Selon un mode de mise en œuvre de l'invention, au cours de l'exécution du fichier exécutable, on contrôle et modifie les chemins d'accès dans le fichier de référence au moyen d'un canal de commande.

On peut également utiliser le canal de commande pour contrôler les fonctions comprises dans le fichier de services.

Les fichiers exécutables sont désormais mutables, sans qu'il soit nécessaire de les régénérer. Des services peuvent être insérés, supprimés, ou modifiés en cours d'exécution.

Le fichier de référence se sert des fonctions d'édition dynamique de liens du système d'exploitation de l'ordinateur pour modifier les fichiers exécutables et activer ou désactiver des services.

Selon l'état de la technique, le processus d'édition dynamique de liens est normalement activée uniquement au chargement du fichier exécutable, avant son lancement. Avec le procédé selon l'invention, ce processus est accessible pendant toute la durée d'exécution du fichier exécutable.

On dit que le fichier de références est réentrant car une même fonction peut être altérée par plusieurs services, l'ordre d'exécution de ces services étant défini par ce fichier de références.

Selon un autre aspect de l'invention, il est proposé un système pour modifier de manière non intrusive un fichier exécutable comprenant des références à des fonctions disposées dans au moins une bibliothèque partagée au sein d'un ordinateur. Avantagusement, ce système comprend :

- un fichier de services comportant une pluralité de fonctions destinées à être insérées de façon dynamique dans le fichier exécutable lors d'une phase d'exécution de ce fichier exécutable,
- un fichier de référence comportant une pluralité de références à des fonctions localisées dans ledit fichier de services et dans la bibliothèque partagée, et
- des moyens de commande pour contrôler et commander le fichier de services et le fichier de référence.

D'autres avantages et caractéristiques de l'invention apparaîtront à l'examen de la description détaillée d'un mode de mise en œuvre nullement limitatif, et des dessins annexés sur lesquels :

- la figure 1 est un schéma simplifié illustrant l'organisation physique d'un fichier exécutable utilisant une bibliothèque partagée selon l'art antérieur;

- la figure 2 est un schéma simplifié illustrant l'organisation physique d'un fichier exécutable utilisant plusieurs bibliothèques partagées selon la présente invention; et
- la figure 3 est un organigramme comprenant différentes étapes d'un procédé selon l'invention.

5

Sur la figure 1 est représentée une organisation physique d'un fichier exécutable dynamique E utilisant une bibliothèque partagée B selon l'art antérieur. Le fichier exécutable E est un programme sous forme de code objet dans lequel sont spécifiés des symboles F1, F2 et F3. Ces symboles sont des références à des fonctions F1, F2 et F3 d'usage commun dont les codes objets coF1, coF2 et coF3 sont stockés dans la bibliothèque partagée B. Cette bibliothèque est un fichier séparé non concaténé au programme E. Avant l'exécution des premières instructions codées dans le programme E, une phase de résolution des références aux symboles est effectuée à chaque chargement en mémoire du programme E par un programme d'amorçage (ou bootloader en langue anglaise). Cette phase de résolution des références fait partie d'une phase d'édition de liens qui a pour but de lier chaque référence Fi du programme E à son code objet stocké dans la bibliothèque B.

10

15

La figure 2 illustre un système selon l'invention dans lequel on retrouve les éléments de la figure 1. La figure 3 décrit différentes étapes d'un procédé selon l'invention.

20

On va maintenant décrire un mode de mise en œuvre de l'invention en référence aux figures 2 et 3.

25

On voit sur la figure 2, deux nouveaux fichiers par rapport à la figure 1. On distingue un fichier de références META qui est une bibliothèque partagée spécifique comportant les références F2 et F3. On distingue également le fichier de services S qui est une bibliothèque partagée spécifique comportant des codes objets coX1, coX2 et coX3 des nouvelles fonctions ou services X1, X2 et X3. On désire modifier l'exécution du

30

programme E en intégrant quelques nouvelles fonctions Xi sans pour cela modifier physiquement le fichier exécutable E et la bibliothèque B.

Dans le fichier META, à chaque référence est associée une équation telle que :

5

$$F2 = S : X2, B : F2$$

et

$$F3 = S : X3$$

10 L'équation associée à F2 signifie que l'on remplace l'exécution de la fonction F2 par l'exécution de la fonction X2 stockée dans le fichier S puis par l'exécution de la fonction F2 stockée dans la bibliothèque partagée B.

L'équation associée à F3 signifie que l'on remplace l'exécution de la fonction F3 par l'exécution de la fonction X3 stockée dans le fichier S.

15 Lors du lancement du programme E à l'étape 1, le système d'exploitation dans lequel opère l'invention est instruit de pré-charger le fichier META à l'étape 2 avant exécution des premières instructions codées dans E. Par exemple, ceci est obtenu sur le système LINUX en positionnant la variable d'environnement "LD\_PRELOAD".

20 Ensuite à l'étape 3 débute la phase d'édition dynamique de liens qui est effectuée par le programme d'amorçage (bootloader). On considère à l'étape 4 toutes les fonctions dont le code objet n'est pas inclus dans le fichier E, c'est-à-dire que le fichier E ne comprend que leurs références ou symboles, on dit alors que les symboles sont non résolus. Sur la figure 2,

25 ces références sont F1, F2 et F3. Elles sont affectées en priorité à META de telle sorte qu'à l'étape 5 on vérifie d'abord si elles sont répertoriées dans le fichier META. Seules les références F2 et F3 sont répertoriées dans le fichier META. Dans ce cas, on édite à l'étape 7 un lien entre les références F1 et F2 du fichier exécutable E et les références F1 et F2 du

30 fichier de références META. Par contre si une fonction n'est pas référencée dans le fichier META telle que la fonction F1, on édite à l'étape 6 un lien



entre la référence F1 du fichier exécutable E et le code objet coF1 de la fonction F1, ce code objet coF1 étant stocké dans la bibliothèque partagée B.

Une fois la phase d'édition de liens terminée à l'étape 8, on débute  
5 réellement l'exécution du fichier E à l'étape 9. Lors de l'exécution de E, l'initialisation du fichier META est déclenchée par le premier appel à l'une des fonctions interceptées, en l'occurrence la fonction F2 à l'étape 10. Cette initialisation consiste à compléter la phase d'édition de liens des étapes comprises entre 3 et 8. Pour ce faire, on utilise les mêmes routines de manipulation des bibliothèques partagées que celles utilisées par le programme d'amorçage. Pour les deux fonctions F1 et F2, sont résolues :

- la référence de chaque fonction dans META vers le service à insérer;  
Sur la figure 2, F2 dans le fichier META doit pointer sur X2 dans le  
15 fichier de services S; F3 dans le fichier META doit pointer sur X3 dans le fichier de services S; En l'absence d'un service à insérer, la fonction est redirigée vers sa bibliothèque d'origine B;
- éventuellement, dans le fichier de services S, la ou les références vers des fonctions pré requises; Sur la figure 2 par exemple, la  
20 fonction x2 dans S rappelle la fonction F2 dans B; Cette action est très importante car elle permet d'établir un chaînage des services, et aussi de pouvoir développer des services incrémentaux.

Cette phase d'initialisation est réalisée aux étapes 11 et 14. Suite à  
cette initialisation de META, le graphe d'appel entre les fonctions est  
25 désormais établi, de sorte qu'on exécute ensuite la fonction X2 à l'étape 12 puis la fonction F2 à l'étape 13.

La capacité de pouvoir redéfinir le graphe d'appel des fonctions est mise en oeuvre au lancement du programme E, de manière à insérer des services sous forme de bibliothèque partagée, mais on peut aussi activer  
30 ces services tout au long de l'exécution de ce programme E, notamment au moyen d'un canal de commande dans le fichier META. L'ouverture de ce

canal de commande peut être déclenchée suivant de multiples stratégies :  
à l'interception de fonctions dans META, sur signal, etc.

A titre d'exemple, on peut utiliser le fichier META pour spécifier un appel système de la façon suivante :

```

5  "
    extern int open(const char *pathname, int flags, mode_t mode)
    {
    if (traceon && tracefile)
        fprintf(tracefile, "trace%s: %d: open(\"%s\", %o, %o)\n",
10         INDEX, pid, pathname, flags, mode);
    Return((*meta_open)(pathname, flags, mode));}
    "

```

Dans l'exemple ci-dessus, la ligne en gras désigne l'appel à la fonction "open" sous-jacente. La variable "meta\_open" est un pointeur sur  
15 la fonction originale, ou une routine implémentant un autre service de manière identique. Cette fonction, par exemple X2 définie dans S, peut être chargée par META pour une pluralité de fichiers exécutables, sans qu'il y ait besoin de les modifier.

Le fichier META est indépendant des programmes modifiés aussi  
20 bien que des services réalisés. Chaque service, sous forme de bibliothèque partagée, fournit au fichier META les données suivantes :

- la liste des fonctions que le fichier META doit intercepter, avec la correspondance de la fonction interne à appeler, par exemple sur la figure 2, l'appel de la fonction F2 dans E doit déclencher la  
25 fonction X2 dans S;
- la liste des fonctions externes (donc pouvant être interceptées) pré-requises pour la mise en oeuvre du service, c'est cette information qui permet au fichier META de réaliser le chaînage entre services et fonctions;
- 30 - la définition de chacune des fonctions de substitution à activer lors de l'interception, ainsi que des fonctions nouvelles variées; et

- la liste des fonctions nouvelles exportées par le service, dont peuvent se servir les services sous-jacents; par exemple, META exporte des fonctions d'ouverture du canal de commandes de manière à ce que les autres services puissent les utiliser sans avoir à les ré-implémenter.

La présente invention apporte une méthode alternative pour aborder simplement des problèmes généraux, tels que la tolérance aux pannes des applications, la migration des applications, le contrôle externe des entrées-sorties. Elle permet de rendre les fichiers exécutables mutables de façon non intrusive. Ainsi, dans un fichier exécutable dans un système selon l'invention, les fonctions mutables sont celles qui sont définies dans des bibliothèques partagées chargeables dynamiquement, que ce soient des appels systèmes, ou d'autres types de fonctions (mathématiques, ...).

Le procédé selon l'invention se sert des fonctions d'édition de liens dynamique du système d'exploitation afin de manipuler de façon non intrusive des fichiers exécutables et activer/désactiver des services, eux-mêmes sous forme de bibliothèques partagées. Les fonctions d'édition dynamique de liens sont avantageusement accessibles durant toute l'exécution du fichier exécutable. Enfin, le fichier de référence META est réentrant puisque une même fonction peut être altérée par plusieurs services.

Bien sûr, l'invention n'est pas limitée aux exemples qui viennent d'être décrits et de nombreux aménagements peuvent être apportés à ces exemples sans sortir du cadre de l'invention. En particulier on peut concevoir plusieurs fichiers de références de type META et plusieurs fichiers de services S.

## REVENDICATIONS

1. Procédé pour modifier de manière non intrusive un fichier exécutable (E) comprenant des références (F1, F2, F3) à des fonctions disposées dans au moins une bibliothèque partagée (B) au sein d'un ordinateur, caractérisé en ce que :
- au lancement du fichier exécutable (E), on pré-charge un fichier de référence (META) comprenant une pluralité de références (F2, F3) à des fonctions nouvelles (X2, X3) disposées dans un fichier de services (S),
  - lors de la phase d'édition dynamique de liens pendant le lancement du fichier exécutable, pour chaque référence non résolue comprise dans le fichier exécutable, on affecte en priorité ladite référence au fichier de référence (META) dans la mesure où cette référence (F2, F3) est définie dans ce fichier de référence, sinon on affecte cette référence (F1) à la bibliothèque partagée (B),
  - au cours de l'exécution du fichier exécutable, l'appel à une fonction référencée dans le fichier de référence, permet l'exécution d'une fonction nouvelle (X3) disposée dans le fichier de service (S) au lieu de l'exécution de la fonction (F3) réellement appelée, le chemin d'accès à cette fonction nouvelle étant définie dans le fichier de référence.
2. Procédé selon la revendication 1, caractérisé en ce que le chemin d'accès défini dans le fichier de référence permet en outre l'exécution d'une fonction (F2) de la bibliothèque partagée (B).
3. Procédé selon l'une des revendications 1 et 2, caractérisé en ce que lors du premier appel à une fonction référencée dans le fichier de référence, on effectue une nouvelle phase (10, 14) d'édition dynamique

de liens entre les références (F2, F3) comprises dans le fichier de référence (META) et des fonctions (X2, F2, X3) en relation avec ces références (F2, F3).

5 4. Procédé selon la revendication 3, caractérisé en ce qu'on effectue la nouvelle phase (10, 14) d'édition dynamique de liens en utilisant des routines de manipulation de bibliothèques partagées.

10 5. Procédé selon l'une quelconque des revendications précédentes, caractérisé en ce qu'on gère le fichier de référence (META) et le fichier de services (S) en tant que des bibliothèques partagées.

15 6. Procédé selon l'une quelconque des revendications précédentes, caractérisé en ce qu'au cours de l'exécution du fichier exécutable (E), on contrôle et modifie les chemins d'accès dans le fichier de référence au moyen d'un canal de commande.

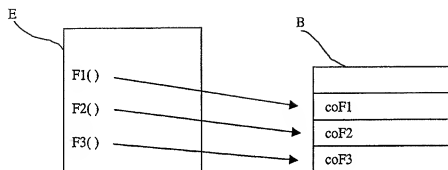
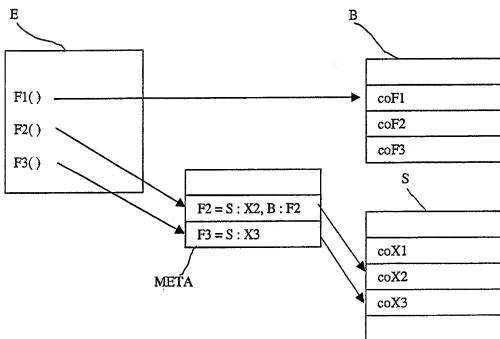
20 7. Procédé selon la revendication 6, caractérisé en ce qu'on utilise le canal de commande pour contrôler les fonctions (X2, X2, X3) comprises dans le fichier de services.

25 8. Système pour modifier de manière non intrusive un fichier exécutable (E) comprenant des références (F1, F2, F3) à des fonctions disposées dans au moins une bibliothèque partagée (B) au sein d'un ordinateur, caractérisé en ce qu'il comprend :

- un fichier de services (S) comportant une pluralité de fonctions (X1, X2, X3) destinées à être insérées de façon dynamique dans le fichier exécutable (E) lors d'une phase d'exécution de ce fichier exécutable,

- un fichier de référence (META) comportant une pluralité de références (F2, F3) à des fonctions localisées dans ledit fichier de services (S) et dans la bibliothèque partagée (B), et
  - des moyens de commande pour contrôler et commander le
- 5      fichier de services (S) et le fichier de référence (META).

1/2

**FIGURE 1****FIGURE 2**

2/2

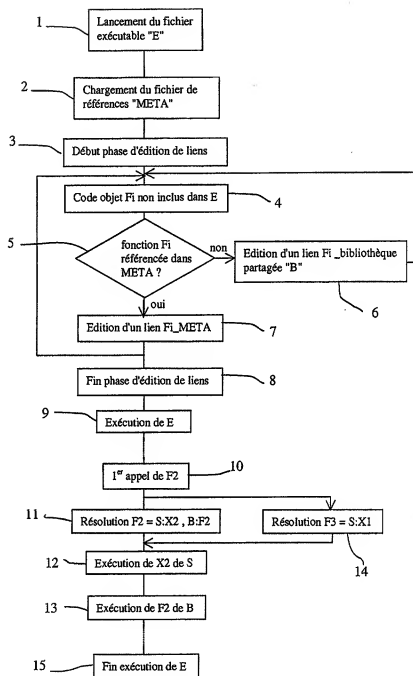


FIGURE 3